Определение остатков при различных видах CRC-контроля

П. Н. Ерлыков

Петербургский государственный университет путей сообщения Императора Александра I

petrerlikov@mail.ru

Н. С. Ерлыков

Петербургский государственный университет путей сообщения Императора Александра I

nikolaysergeevich.yerlikov@mail.ru

Ю. Я. Меремсон

Петербургский государственный университет путей сообщения Императора Александра I

meremson@list.ru

Аннотация. **Авторами рассмотрена логика работы с «выровненным» сообщением и показана возможность задавать на передающем пункте и возвращать в пункт приема любые остатки, разрядность которых равна ширине полинома, что расширяет возможности CRC-контроля.**

Ключевые слова: CRC-алгоритм, выровненное сообщение, полином, остаток, делимое, делитель, займ

I. Постановка задачи

В работе [1] приведены полезные для применения материалы по использованию CRC-контроля, однако объяснения самих основ применения алгоритмов CRC авторов не удовлетворили.

В первоисточнике [2] (страница 238) единственного примера выполнения алгоритма CRC показана работа с «выровненным» сообщением, равным двоичному числу, к которому добавлено число нулей, равное степени старшего члена полинома – делителя.

Авторы данной работы решили, что прежде, чем браться за объяснение работы алгоритма CRC над «выровненным» сообщением, следует объяснить более простое действие алгоритма CRC над обычным сообщением без добавленных нулей, тем более что вполне доступен такой вариант выполнения контроля CRC.

Для определения остатков при двух видах CRC-контроля: без добавления нулей и при добавлении нулей будем применять в принципе тот же способ, который был использован нами в работе [3].

Забегая вперёд, заметим, что от первой таблицы в каждой из групп оставлена лишь запись величины «нормального» остатка от контролируемого числа без добавления нулей.

Число групп примем равным возможному числу контрольных чисел (остатков) от применения алгоритма CRC с полиномом 10011 (19).

Число остатков при обычном делении чисел на делитель, равный 19-ти (с учётом деления без остатка) равно, естественно, 19-ти.

Однако, как мы уже уяснили, алгоритм CRC с двоичными числами отличается от обычного деления тем, что вычитания по таблице Горнера выполняются не обычно, а поразрядно, с отбрасыванием займов.

Это приводит, в частности, к тому, что двоичная разрядность «нормальных» остатков становится равной ширине полинома. А так как в нашем случае использован полином с шириной, равной 4-м, то «нормальный» остаток может представлять собой любое число от числа 0000 (0) до числа 1111 (15), то есть возможное число остатков (с учётом деления без остатка) равно 16-ти. В скобках, как обычно, приведены десятичные эквиваленты двоичных чисел.

В работе [3] три числа для трёх групп таблиц (это числа 561, 566, 565) выбраны так, чтобы показать закономерность работы алгоритма CRC как над числами без добавленных нулей, так и над числами с добавленными нулями, а также над теми числами, у которых вместо добавленных нулей появились остатки.

Применением трёх исходных чисел мы, в частности, показали работу алгоритма при различных «средних» остатках, образующихся при передаче числа в пункт приёма вместо добавленных четырёх младших нулей («специального» остатка).

Эти остатки равны: 0001 (1) для числа 561, 1000 (8) – для числа 566 и 1101 (13) для числа 565.

Для получения таких разных «нормальных остатков нам достаточно было в качестве исходных взять: во-первых, изначально примененное нами число 561, а также числа 566 и 565, немного отличные от изначального.

Теперь же нам нужно для каждой из трёх групп таблиц, как и в работе [3] показать работу алгоритма CRC при обработке трёх чисел:

- первого – исходного числа без добавленных четырёх нулей;
- второго – того же контролируемого числа с добавленными четырьмя нулями;
- третьего – того же контролируемого числа с полученным при обработке второго числа «средним» остатком вместо четырёх нулей.

Из-за применения в алгоритме CRC поразрядного вычитания вместо обычного вычитания, для выбранных по натуральному ряду контролируемых чисел «средние» остатки не следуют тем же рядом, а следуют закономерностью, зависящей как от исходных чисел, так и от принятого полинома.

Для выполнения нашей задачи можно в качестве контролируемых чисел (без добавления нулей) брать непосредственно сами возможные остатки.

Рассматривать эти остатки удобнее, если последовательность задать в соответствии с натуральным рядом.

Начинать этот ряд можно с исходного контролируемого числа, равной максимальной величине остатка: 1111 (15), затем брать исходное число, меньшее на единицу и т.д.

Рассмотрим первую группу с исходным остатком, равным числу 1111 (15).

Замечаем, что исходным делимым в пункте передачи является двоичное число 11110000. А первым уменьшаемым для поразрядного вычитания полинома 10011 является число 11110.

В результате вычитания получается число 1101, добавляя которое нулём, получаем второе уменьшаемое 11010.

Выполняя второе поразрядное вычитание полинома, получаем в результате число 1001, добавляя которое одним нулём получаем третье уменьшаемое 10010, вычитая из которого поразрядно полином, получаем число 001, добавляя которое самым младшим нулём получаем остаток, равный числу 0010, то есть равный двум.

Таким образом, мы получили однозначное соответствие между двумя остатками «нормальным» остатком 1111 при исходном числе без добавления четырёх нулей и «средним» остатком при том же исходном числе с добавленными четырьмя нулями, равном числу 0010.

Теперь исходное число с полученным на пункте передачи «специальным» остатком (то есть число 11110010) обрабатывается алгоритмом CRC.

Первым уменьшаемым для поразрядного вычитания полинома является число 1101, добавляя которое нулём, получаем второе уменьшаемое 11010.

После второго поразрядного вычитания полинома получаем число 1001, добавляя которое единицей нового остатка, получаем третье уменьшаемое 10011, равное полиному. Выполняя третье поразрядное вычитание полинома, естественно, получаем нулевой остаток: 0000 (0).

Все последующие группы таблиц, соответствующие остальным начальным контролируемым числам – «нормальным» остаткам от применения алгоритма CRC к исходным числам без добавления нулей можно проследить самостоятельно.

Иными словами, рассмотрим случай, когда для исходного контролируемого числа без добавления нулей в результате применения алгоритма CRC «нормальный» остаток оказался равным нулю.

С точки зрения второго правила выполнения алгоритма CRC, приведённого в [3], выполнять поразрядное вычитание полинома нельзя, так как у уменьшаемого нет единицы ни в одном разряде.

Результат виден и чисто логически, так как деление нуля на любое число даёт нуль для частного и для остатка.

В итоге как «Начальное число», так и «Посылка» и так же «Остаток» в пункте приёма сообщения, естественно равны нулю.

В табл. 1 приведём соответствие «нормальных» (исходных) остатков, которые, как нам известно, получены в результате применения алгоритма CRC к контролируемым числам без добавления четырёх нулей, со «средними» остатками, полученными в результате применения алгоритма CRC к тем же контролируемым числам с добавленными четырьмя нулями

ТАБЛИЦА I.

«Нормальный» остаток при применении алгоритма CRC к числам без добавленных 4-х младших нулей	**«Средний» остаток при применении алгоритма CRC к числам с добавлением 4-х младших нулей**
1111 (15)	0010 (2)
1110 (14)	0001 (1)
1101 (13)	0100 (4)
1100 (12)	0111 (7)
1011 (11)	1110 (14)
1010 (10)	1101 (13)
1001 (9)	1000 (8)
1000 (8)	1011 (11)
0111 (7)	1001 (9)
0110 (6)	1010 (10)
0101 (5)	1111 (15)
0100 (4)	1100 (12)
0011 (3)	0101 (5)
0010 (2)	0110 (6)
0001 (1)	0011 (3)
0000 (0)	0000 (0)

В табл. 1 приведена двоичная форма чисел остатков, десятичная форма тех же чисел приведена в скобках.

Следование чисел «нормальных» остатков в столбце справа по порядку друг за другом от 15-ти до нуля соответствует следованию этих чисел в порядке расположения групп таблиц.

Беспорядочное следование «средних» остатков в столбце справа вполне объяснимо принципом поразрядного вычитания с отбрасыванием займов, заложенным в алгоритме CRC.

Естественно, если расставить по порядку числа столбца справа, то в беспорядке расположатся числа в столбце слева.

Табл. 1 фактически имеет характер справки, которая может понадобиться при применении способа контроля, приведенного в работе [3].

Обратим ещё раз внимание на цели использования чисел с добавленными нулями, число которых принимают равным ширине используемого полинома.

II. Выводы

Автор работы [1] указывает, что целью добавления нулей является продление работы алгоритма с полиномом для обработки «хвостовой» части полинома.

Но главное – нужна ли такая обработка? На этот вопрос у автора работы [1] нет ответа.

В отношении такой обработки, имеющей как плюсы, так и минусы можно провести отдельное обсуждение, выходящее за рамки нашей статьи.

В то же время ясна выгода добавления передаваемых чисел нулями. Нули добавляются на передающем пункте в процессе передачи сообщения на приёмный пункт при обработке передаваемого сообщения алгоритмом CRC, и «специальный» остаток с разрядностью, равной количеству добавленных нулей и замещающий эти нули, возникает в том же процессе передачи сообщения на приёмный пункт. Именно это упрощение по сравнению с начала контролируемого числа, а затем остатка, и отдельное сравнивание остатков, как выполнялось бы по описанию [3], является, по нашему мнению, очевидным преимуществом применения работы с «выровненным сообщением».

Список литературы

[1] Ross N. Williams. Элементарное руководство по CRC-алгоритмам обнаружения ошибок. Текст электронный. https://www.studmed.ru/ross-williams-n-elementarnoe-rukovodstvo-po-crc-algoritmam-obnaruzheniya-oshibok_9ab773e8cd3.html

[2] Таненбаум Э., Уэзеролл Д. Т.18 Компьютерные сети. 5-е изд. СПб.: Питер. 2012. 960 с.: ил.

[3] Привалов А.А., Ерлыков П.Н., Ерлыков Н.С. Методы применения CRC-контроля для помехозащиты телекоммуникационных систем // ELTRANS-2023. Сборник трудов XI международного симпозиума. ПГУПС. 2023. С. 392-397.

[4] Глоссарий простых телекоммуникационных терминов (ГПТТ). Текст электронный. https://www.itu.int/rec/T - REC-G.704/en. Версия 17.02.2021 15.29.